METHOD AND SYSTEM FOR DE-JITTERING TRANSMITTED MPEG-2 AND MPEG-4 VIDEO

Cross-Reference to Related Applications

This application claims priority under 35 U.S.C. § 119 based on U.S. Provisional Application Serial No. 60/167,339, filed November 24, 1999, the disclosure of which is incorporated by reference.

Statement Regarding Federally Sponsored Research Or Development

This invention was made with Government support under Contract No. DAAL-01-96-2-0002, awarded by the U.S. Army Research Laboratory. The Government has certain rights in this invention.

Field of Invention

This invention relates generally to the field of multimedia transmission over a network. More specifically, this invention relates to method of de-jittering MPEG-2 and MPEG-4 video data transmitted over a packet switched network.

Background of the Invention

The MPEG-2 and MPEG-4 standards are well-known in the art for coding and storing multimedia video and associated audio information. When MPEG multimedia data is transmitted over a network from a source device to a destination device, it is important that the transmitted data be synchronized at the destination device by matching the destination device's clock to the source device's clock. It is known in the art to use a phase locked loop (PLL) at the destination device to synchronize the source device's clock with the destination device's clock.

Generally, as is known in the art, MPEG-2 and MPEG-4 standards call for multimedia data to be coded and stored in discrete data packets. The format of each data packet provides for a "clock-stamp" reference value in which a time reference value from the source device's clock can be stored prior to transmission across the network. When a stream of data packets are transmitted over a network, only a selected sample of the data packets actually include a clock-stamp time reference stored in the reserved data bytes. The destination device compares the clock-stamp time references that it receives in the transmitted MPEG data with the instant time provided by the destination device's local clock. From this comparison, a phase error can be derived. A PLL uses the phase error to adjust the decoder clock. Methods of comparing clock-stamp time references with the destination device's clock to determine a phase error and enable a PLL to adjust the destination device's clock to match the source device's clock are known in the art.

For purposes of synchronizing the device's respective clocks, MPEG semantics assume a constant delay network between the source device and the destination device. However, it is difficult, if not impossible, to maintain a constant network delay. Non-constant network delays, known as "jitter", can result in a degradation of the video playback. Jitter results in data packets arriving at the destination device in a non-uniform manner, which impedes effective clock synchronization by the PLL. Specifically, the PLL must perform additional filtering in order to correctly estimate the STC clock values. This, in turn, slows down the responsiveness of the PLL and affects the maximum phase error introduced by the PLL between the clock-stamped reference values encoded from the source device's clock and the corresponding destination device's time clock references. To assure a stable recovery of the source device's clock values (also referred to as the system clock (STC)) by the PLL, de-jittering algorithms must be performed before the encoded clock values are passed to the PLL.

Brief Summary of the Invention

The present invention comprises an improved method and system for reducing jitter in MPEG data transmissions due to non-constant network delay times. Generally, the present invention calculates a statistical estimation of the average network system jitter. The estimated average network system jitter is then used to re-calculate a "corrected" reference value for subsequent clock-stamp reference values. Specifically, for each data packet that contains a clock-stamp reference value, the clock-stamp reference value is parsed out from the rest of the data packet. The average network jitter is estimated based on a prior pre-determined sample of data packets. An estimated jitter is then calculated for the reference data packet. The estimated reference jitter is then translated to clock tics and a "corrected" clock-stamp reference value is calculated. Finally, the original clock-stamp reference value of the subsequent reference data packet is replaced with the "corrected" clock reference value, which includes compensation for the statistical estimation of network jitter, before it is sent to a phase locked loop (PLL). Since the new clock reference values are "corrected" based upon the statistical estimation of the average network system jitter, the phase error of the PLL is minimized, resulting in a more stable system time clock (STC). Among other benefits, the present invention improves the quality of the received video and enables the system to tolerate more network jitter without video degradation.

Description of the Drawings

Figure 1 shows a representative MPEG data packet comprising a header portion and a payload portion.

Figure 2 is a flowchart illustrating the steps of the present invention.

Figure 3 comprises timing diagrams that illustrate the relative times between the transmission and receipt of MPEG data packets.

Detailed Description of a Preferred Embodiment

MPEG-2 and MPEG-4 video standards provide for multimedia data to be coded and transported in data packets. As shown in Figure 1, each MPEG-2 and MPEG-4 data packet comprises a header portion and a payload portion. As is known in the art, the header portion of the packet contains administrative information about the data packet, such as packet ID, transport priority, etc. The payload portion of the packet contains video and audio data. Depending on the format of the data packets (either MPEG-2 or MPEG-4), each header portion contains a Program Clock Reference (PCR) or Object Clock Reference (OCR), both of which correspond to the source device's clock at the time the reference data packet is transmitted. PCR or OCR data is included periodically in data packets transmitted from the source device to the destination device, and the data is used to synchronize the system clock reference (STC) at the destination device with the clock at the source device.

Figure 2 shows a flow-chart that illustrates the steps of the present invention, and Figure 3 shows a source device time line and a destination device time line that illustrates the relative timing of data packets transmitted from a source device to a destination device. The various time intervals shown in Figure 3 assist in illustrating the steps shown in Figure 2.

Referring to Figures 2 and 3, it is assumed that the source device transmits a stream of MPEG data packets with a constant nominal period T -- i.e., with a constant nominal time period between each data packet transmission. The particular value of period T depends upon the application, and the present invention can be used in connection with any period T. As shown in Figure 3, data packets are transmitted with period T from the source device at departure times (D_t). Arrival times A_t correspond to theoretical arrival times at the destination device, assuming a constant delay network with delay time D_{ref}. However, because the network has a non-constant delay time, the actual arrival times differ from the theoretical arrival times. The actual arrival times are designated in Figure 3 as A_a. For each data packet that arrives at the destination device, the differences between the actual arrival time A_a and the theoretical arrival time A_t constitutes the jitter J for that particular data packet. Each data packet that arrives at the destination device is stored in a computer memory device that is of the type that is well-known in the art.

It is assumed that a clock-stamp reference value carrying a snap shot of the value of the clock at the source device is periodically stored in the header portion of data packets and sent every T_{ref} time, or every N packets. Again, the particular frequency with which reference clock values are inserted -- the particular values of T_{ref} and/or N -- do not affect the applicability of the present invention. The value of T_{ref} could be an inter-PCR or inter-OCR period, as is known in the art, depending upon the specific transport mechanism used. Each data packet that contains a clock-stamped reference value is considered a reference data packet.

In step 22 of Figure 2, the destination device's clock is initiated to the clock-stamp reference value of the first data packet received by the destination device that includes a clock-stamp reference value. The initiation of the destination device's clock in this manner is done by default as per the MPEG standards. In step 26, a destination device counter is also initiated to the reference clock value to be used in connection with de-jittering. The destination device counter is used to register the actual arrival times of the incoming MPEG data packets.

Per step 28 of Figure 2, the arrival time of the first reference data packet carrying a clock-stamp reference value is registered and saved. Then, as shown in step 30, the actual arrival times (A_a) of all subsequent packets (N) received between two successive reference data packets are registered and saved. The arrival times are stored in computer memory devices that are well-known to those skilled in the art. The actual arrival times (A_a) of the N packets are referred to herein as A_{ai}, where i=1 to N. In step 34, the theoretical arrival times (A_t), assuming a constant delay network, of the N packets are calculated using the actual arrival time of the reference data packet (A_{refl}) as a reference point. Specifically, the theoretical arrival times (A_{ti}, where i=1 to N) are calculated as follows:

$$A_{ti} = A_{arefl} + i * T,$$

where A_{aref1} represents the actual arrival time of the most currently-received reference data packet. As shown in step 38 of Figure 2, a jitter value (J_i, where i=1 to N) is calculated for each received data packet by subtracting the actual arrival times (A_a) from the theoretical arrival times (A_t) according to the following formula:

$$J_i = A_{ti} - A_{ai}.$$

After all of the jitter values (J_i) have been calculated for the current subset of N data packets, a sample mean jitter (μ) is calculated, as shown in step 42, according to the following formula:

$$\mu = (1/N) * \sum_{i=1}^{N} J_i$$

The calculated sample mean jitter value (μ) can be positive, negative, or zero depending on the delay (D_{ref}) experienced by the reference data packet and the number of data packets (N) in the sample subset. The sample mean jitter (μ) represents the average network system jitter over the current sample of N data packets.

Based upon the calculated sample mean jitter value (μ), the jitter of the next reference data packet is estimated. Specifically, as shown in step 46, a "corrected" theoretical arrival time (A_{ctref2}) is calculated for the next reference data packet according to the following formula:

$$A_{ctref2} = (A_{aref1} + (N+1) * T) - \mu$$

According to the above formula, the corrected theoretical arrival time of the next reference data packet (A_{ctref2}) is determined by calculating the uncorrected theoretical arrival time ($A_{aref1} + (N+1) * T$) and subtracting the estimated mean network jitter (μ).

The corrected theoretical arrival time of the next reference data packet (A_{ctref2}) is used to calculate the jitter associated with that data packet (J_{ref2}) . After the next reference data packet containing a clock-stamped reference value is received (step 48), the jitter of that reference data packet is calculated by subtracting the actual arrival time from the corrected theoretical arrival time according to the following formula, as shown in step 50:

$$J_{ref2} = A_{ctref2} - A_{aref2},$$

where A_{ctref2} is the corrected theoretical arrival time of the next reference data packet and A_{aref2} is the actual arrival time of the next reference data packet. The corrected theoretical arrival times and the jitter values of the clock-stamp reference values are determined by an electronic controller that is of the type that is well-known in the art.

The corrected theoretical arrival time of the newly-received reference data packet is then used as a reference point for the calculation of the sample mean jitter of the next N data packets. Specifically, the sample mean jitter of the next N data packets is calculated as described above, except that the corrected theoretical arrival reference time (A_{ctref2}) replaces the actual arrival time reference (A_{aref1}) described hereinabove. Since the jitter calculation of the next N packets is based on a clock-stamped reference time that incorporates compensation for an estimated average network delay, the value of μ for the following sets of N data packets should be close to zero and exhibit little variation under the same network operating conditions.

In step 54, the jitter value (J_{ref2}) is translated to an adjustment step (Δ) in terms of the number of STC tics, according to the following formula:

$$\Delta = J_{ref2} * \text{STC resolution},$$

where J_{ref2} is measured in seconds, and STC resolution is in tics per second. Based on the Δ value, a corrected clock-stamp reference value is calculated. As shown in step 58, the corrected clock-stamp reference value, which includes compensation for the average network delay, replaces the actual clock-stamp reference value stored in the reference data packet before it is sent to the PLL. Replacing the received clock-stamped time reference with the calculated corrected clock-stamp time reference before it is sent to the PLL minimizes the phase error of the PLL and provides a more stable STC reconstruction.

The above-described process is repeated, as shown in Figure 2, each time a new reference data packet having a clock-stamp reference time included therein is received by the destination device. In this way, the actual clock-stamp reference value of each reference packet is replaced with a corrected clock-stamp reference value that incorporates compensation for the network system jitter. As a result, the destination device's PLL is more effective in recovering the system clock STC, which improves the quality of the video playback at the destination device.

While a preferred embodiment of the present invention has been described herein, it is apparent that the basic construction can be altered to provide other embodiments that utilize the processes and compositions of this invention. Therefore, it will be appreciated that the scope of this invention is to be defined by the claims appended hereto rather than by the specific embodiment that has been presented hereinbefore by way of example.